

DEVICE BOUND FLASHING/BOOTING FOR CLONING PREVENTION

PRIORITY CLAIM

[0001] This application claims priority to U.S. Provisional Patent Application Serial No. 60/510,696, filed on October 10, 2003, entitled "DEVICE BOUND FLASHING/BOOTING FOR CLONING PREVENTION," incorporated herein by reference.

BACKGROUND

[0002] Mobile phones generally comprise software applications that may be executed to operate the mobile phone. In addition to enabling a phone with voice communications capabilities, these software applications may enable a phone with various other capabilities, such as text messaging and digital photography. A mobile phone boot image may comprise an operating system and any of a variety of such software applications that may be executed on the mobile phone. The price of a mobile phone may vary based on the quality of the boot image embedded in the phone's flash memory. High-quality boot images may cause particular phones to be more expensive than phones with boot images of lesser quality.

[0003] Texas Instruments[®] proprietary Open Multimedia Applications Platform ("OMAP") comprises a microprocessing engine that enables communications devices to process data and software applications while extending battery life. A mechanism present in current OMAP devices (i.e., models 161x, 171x, 73x) supports the flashing and booting of boot images using a key that is shared among a plurality of devices. This key helps verify the authenticity of a boot image, but does not prevent the unauthorized copying and re-use of the boot image on a separate phone, resulting in a possible penetrable security gap. Such a security gap may enable unauthorized entities to copy boot images from expensive phones and reproduce the boot images on inexpensive phones. In this way, an unauthorized entity may clone an expensive phone into an unlimited number of inexpensive phones and sell the inexpensive phones for a profit.

[0004] In addition to unlawfully copying the boot image, unauthorized entities also may tamper with the contents of the boot image to circumvent existing safeguards that prevent the usage of stolen mobile phones. For example, each mobile phone boot image comprises an International Mobile Equipment Identifier ("IMEI") number that serves as an identification code for the phone in the Global System for Mobile Communication ("GSM") and Third Generation ("3G") networks. The IMEI number is used to grant or deny access to the cellular networks and the networks' services. Generally, if a phone is stolen, the owner may contact his or her cellular service provider (e.g., Sprint®, Verizon®, AT&T®) and have the phone added to a GSM/3G blacklist. Mobile phones found on the blacklist will be denied access to the cellular networks. Thus, an unauthorized entity that steals the phone would not be able to use the phone to access the networks, because the IMEI number of the phone has been added to the blacklist. However, a knowledgeable, unauthorized entity may easily alter the IMEI number of the stolen phone to a number that is not found on the blacklist, thereby gaining access to the cellular networks by way of the stolen phone.

[0005] Each year, mobile phone manufacturers lose substantial amounts of revenue due to phone cloning and tampering. Thus, it is desirable to prevent phone cloning and tampering.

BRIEF SUMMARY

[0006] The problems noted above are solved in large part by a method and apparatus for binding a boot image and the various contents of a boot image to a mobile communication device. One exemplary embodiment may include downloading a boot image onto a mobile communication device and generating a device-bound certificate ("DBC"). The DBC preferably comprises an authentication code generated using a hashed message authentication code ("HMAC") algorithm and a key specific to the device. The method may further comprise storing the DBC on the boot image, thus binding the boot image to the mobile communication device.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] For a detailed description of exemplary embodiments of the invention, reference will now be made to the accompanying drawings in which:

[0008] Figure 1 illustrates a block diagram of a mobile communication device in accordance with embodiments of the invention;

[0009] Figure 2 illustrates an exemplary process implemented on the mobile communication device of Figure 1;

[0010] Figure 3a illustrates a block diagram of a flashing-process boot image in accordance with a preferred embodiment of the invention;

[0011] Figure 3b illustrates a block diagram of a booting-process boot image in accordance with a preferred embodiment of the invention; and

[0012] Figure 4 illustrates a flow diagram of a device bound certificate (“DBC”) authorization process in accordance with a preferred embodiment of the invention.

NOTATION AND NOMENCLATURE

[0013] Certain terms are used throughout the following description and claims to refer to particular system components. As one skilled in the art will appreciate, various companies may refer to a component by different names. This document does not intend to distinguish between components that differ in name but not function. In the following discussion and in the claims, the terms “including” and “comprising” are used in an open-ended fashion, and thus should be interpreted to mean “including, but not limited to... .” Also, the term “couple” or “couples” is intended to mean either an indirect or direct electrical connection. Thus, if a first device couples to a second device, that connection may be through a direct electrical connection, or through an indirect electrical connection via other devices and connections.

DETAILED DESCRIPTION

[0014] The following discussion is directed to various embodiments of the invention. Although one or more of these embodiments may be preferred, the embodiments disclosed should not be interpreted, or otherwise used, as limiting the scope of the disclosure, including the claims. In addition, one skilled in the art will understand that the following description has broad application, and the discussion of any embodiment is meant only to be exemplary of that embodiment, and not intended to intimate that the scope of the disclosure, including the claims, is limited to that embodiment.

[0015] In accordance with the preferred embodiments, per-device “binding” of boot images and boot image contents is provided for a mobile communication device. A boot

image that has been downloaded to a mobile phone may be manipulated so that the boot image cannot be copied, altered, or transferred to any other mobile phone. In this way, the boot image is “bound” to the mobile phone. The boot image, once bound to the mobile phone, is valid only for that particular mobile phone. The contents of a boot image also may be bound to a mobile phone in a similar fashion.

[0016] Per-device binding of a boot image is accomplished by way of a device-bound certificate (“DBC”). In accordance with the preferred embodiments, a DBC is used to bind a boot image to a particular mobile phone so that the boot image cannot be transferred, altered or otherwise copied to another mobile phone. In a binding process, private data comprising a hashed message authentication code (“HMAC”) is stored in the DBC and the DBC subsequently is encrypted with a secret key. In the preferred embodiment, a mobile phone is not permitted to use the boot image without first obtaining the HMAC contained in the DBC. The HMAC thus functions to bind the boot image to the mobile phone. Further, the phone cannot access the HMAC in the DBC without the secret key and only the phone to which the boot image is bound has the secret key. Hence, only the phone with the correct secret key may freely access the contents of the boot image. Thus, per-device binding of a boot image and all contents of the boot image is accomplished by way of a DBC.

[0017] Referring now to Figures 1, 2 and 3a, Figure 1 shows a preferred embodiment of a mobile phone 170 comprising an OMAP processor 172 coupled to a UART/USB port 174, a flash memory 178 and comprising a ROM code (i.e., on-chip firmware) 176. Figure 2 illustrates a flow diagram describing the process of binding a boot image to the mobile phone 170. Figure 3a illustrates a boot image 300 comprising a TOC field 302 that describes the contents of the boot image 300, a KEYS header 304 that comprises keys used for cryptographic reasons as described below and a common header 306 that acts as a header for a flash loader 308, which comprises a protected application (“PA”) 318. The boot image 300 also may comprise other fields 310, a PA 312 and an empty device-bound certificate (“DBC”) field 314. Protected applications are thusly named because the protected applications operate in a secure-mode environment, which may be defined as a hardware-based secure execution environment that is generally tamper-proof.

[0018] A binding process may be performed during manufacture of a phone, after the phone has been sold to a consumer, or at any other time. In general, the binding process begins with the creation of a DBC during the flashing process, the filling of the empty DBC field 314 with this DBC, and the subsequent storing of the boot image 300 on the mobile phone 170. More specifically, the binding process may begin with the authentication of the flash loader 308 by the ROM code 176 to ensure the validity of the flash loader 308 (block 202). Once authenticated, the flash loader 308 downloads the boot image 300 by way of a UART/USB port 174 or any appropriate device (block 204). The boot image 300 and other information may be downloaded by a manufacturer or any appropriate entity from any appropriate source, such as the manufacturer's computer systems. In cases where specific items (e.g., an IMEI certificate comprising an IMEI number; SIMlock files) are to be bound to the mobile phone 170, the items may be downloaded in a manner similar to that used to download the boot image 300. Prior to being downloaded, the IMEI certificate preferably is signed by a manufacturer with an Original Equipment Manufacturer Interface ("OEMI") private key. An OEMI public key and the IMEI certificate are both downloaded onto the mobile phone 170, so that the mobile phone 170 may verify the IMEI certificate using the OEMI public key at a later time.

[0019] The flash loader 308 subsequently may load and call the PA 318 (block 206). When calling the PA 318, the flash loader 308 sends various parameters, comprising pointers to various components of the boot image 300 (e.g., the common header 306) as well as the values of Creator ID and Application ID found in the common header 306. The Creator ID describes the owner or creator of a DBC and the Application ID serves as an identifier for the application that creates the DBC. The PA 318 may use these pointers and values as necessary.

[0020] At least one purpose of the PA 318 is to compute the DBC (block 208), optionally encrypt the DBC with a random key (block 210), and pass the DBC to the flash loader 308 for further processing (block 212). As previously discussed, the PA 318 operates in a secure-mode environment. The PA 318 may begin generating the DBC as follows:

$$\text{HMAC} = \text{HMAC}_{\text{KEY}}(\text{SHA-1}(\text{Common Header 306} + \text{Boot Loader}) \parallel \text{Public Chip ID} \parallel \text{Creator ID} \parallel \text{Application ID} \parallel \text{Reserved Fields}),$$

where “HMAC” denotes a hashed message authorization code, the symbol “||” denotes concatenation, the Public Chip ID serves as a public identifier for the OMAP PROCESSOR 172, the Reserved Fields contain any information (e.g., an IMEI certificate) and the boot loader is contained in the boot image 300 as described in Figure 3b below. Specifically, the common header 306 and the boot loader are first hashed together using the commonly-known SHA-1 algorithm, described below. The result is concatenated with various data as shown above (e.g., Public Chip ID, Creator ID). The resulting concatenation is hashed using a key (i.e., KEY) by a commonly-known HMAC cryptographic algorithm, where KEY is generated as:

$$\text{KEY} = \text{SHA-1}(\text{Chip Specific ID} \parallel \text{Creator ID} \parallel \text{Application ID}),$$

and where the Chip Specific ID is a secret identifier created by the ROM code 176 or other system firmware and available only inside secure mode (i.e., during the execution of a PA). A secure hash algorithm SHA-1 is used for computing a “condensed representation” of a message or a data file. The “condensed representation” is of fixed length and is known as a “message digest” or “fingerprint.” It is computationally infeasible to produce two messages having the same message digest. This uniqueness enables the message digest to act as a “fingerprint” of the message. For instance, SHA-1 may be used to ensure the integrity of a downloaded or received file by comparing the file hash with the original file hash. Any message or similar construct requiring integrity may be verified in this fashion.

[0021] The PA 318 completes the DBC computation by assembling a DBC as illustrated in Figure 3b using information computed by the PA 318 or received from the flash loader 308. Specifically, the completed DBC may comprise a Public Chip ID 322, a Creator ID 324, an Application ID 326, a boot loader/common header hash 328, reserved fields 330 and an HMAC 332 generated as described above. The reserved fields 330 may be filled with an IMEI certificate 330 if an IMEI certificate was downloaded in block 204. The reserved fields 330 also may be filled with any other device-specific information. The PA 318 then may optionally encrypt the DBC with a random, secret key K, computed as:

$$K = \text{SHA-1}(\text{Chip Specific ID} \parallel \text{Creator ID} \parallel \text{Application ID}).$$

Encrypting the DBC with a random, secret key K protects all of the contents of the DBC (e.g., the IMEI certificate 330). Once the DBC is encrypted or the encryption step is bypassed, the PA 318 passes the DBC to the flash loader 308 for further processing.

[0022] The flash loader 308 receives the DBC from the PA 318 and inserts the DBC into the empty DBC field 314 (block 214), thereby establishing a DBC 314 inside the boot image 300. The flash loader 308 then completes the binding process by flashing (i.e., writing) the boot image 300 to the flash memory 178 of the mobile phone 170 (block 216).

[0023] The boot image 300 comprising the DBC 314 and bound to the phone 170 cannot be used until the DBC 314 is authenticated at boot time (i.e., each time the phone is turned on) as illustrated in Figure 4. That is, the operating system contained in the boot image 300 will not load unless the DBC 314 is first authenticated, thus preventing an end-user from using the phone 170.

[0024] Referring now to Figures 3b and 4, Figure 3b illustrates a booting-time boot image 300 comprising a TOC field 302, a KEYS header field 304, a common header 306, a boot loader 308 comprising a PA 320, other fields 310, a PA 312 and a DBC 314. As described above, the DBC 314 comprises a Public Chip ID 322, a Creator ID 324, an Application ID 326, a boot loader/common header hash 328, reserved fields 330 that may comprise an IMEI certificate 330, and the HMAC 332. Figure 4 illustrates a flow diagram of the process for authenticating a boot image 300 at boot-time. Specifically, the boot-time authentication process may begin with the verification of the KEYS header 304 and the common header 306 by the on-chip ROM code 176 (block 402). The boot loader 318 may load and call the PA 320 using various parameters comprising pointers to the DBC 314, the common header 306 and boot loader 318, and values of the Creator ID 324 and Application ID 326 from the common header 306 (block 404) so the PA 320 may use these pointers and values as necessary.

[0025] The PA 320 then may verify the integrity of the DBC 314 and, if applicable, the IMEI certificate 330 (block 406) by first unlocking (i.e., decrypting) the DBC 314 (if the DBC 314 was encrypted) using a key K1 and verifying the IMEI certificate 330 using the

OEM public key that was downloaded onto the mobile phone 170 concurrently with the IMEI certificate 330. The key K1 is computed as follows:

$$K1 = \text{SHA-1}(\text{Chip Specific ID} \parallel \text{Creator ID} \parallel \text{Application ID}).$$

Although computed separately, the key K1 used to decrypt the DBC 314 during the booting process is identical to the key K used to encrypt the DBC 314 during the flashing process. After the encrypted DBC 314 is unlocked (if applicable), the PA 320 computes:

$$\text{HMAC1} = \text{HMAC}_{\text{KEY1}}(\text{SHA-1}(\text{Common Header 306} + \text{Boot Loader 318}) \parallel \text{Public Chip ID 322} \parallel \text{Creator ID 324} \parallel \text{Application ID 326} \parallel \text{Reserved Fields}),$$

where the Creator ID 324 and the Application ID 326 are obtained from the DBC 314 and where KEY1 is computed as:

$$\text{KEY1} = \text{SHA-1}(\text{Chip Specific ID} \parallel \text{Creator ID} \parallel \text{Application ID}).$$

The PA 320 subsequently compares the HMAC1 calculated above to the HMAC stored in the DBC 314 to test for a match and passes the result of the comparison to the boot loader 318 (block 408). A match indicates that the boot image 300 has not been copied or altered and may be used by the mobile phone 170 on which the boot image 300 is located. A match also indicates that the contents of boot image 300 (e.g., the IMEI certificate 330) have not been copied or altered and are authentic. In such a case, the booting process would continue as normal. Conversely, a mismatch indicates that the boot image 300 may have been stolen, altered or copied. Thus, the integrity of the contents of the boot image 300 (e.g., the IMEI certificate 330) may have been compromised. In such a case, the booting process would not continue. The boot loader 318 receives the results of this comparison from the PA 320 and proceeds accordingly (block 410), thereby completing the boot-time authentication process.

[0026] Although the subject matter disclosed herein is described in terms of the OMAP161x platform, the OMAP 73x platform, the OMAP 171x platform or any of a variety

of platforms may be used. The above discussion is meant to be illustrative of the principles and various embodiments of the present invention. While the technique for per-device binding of boot image contents is discussed in context of IMEI certificates, the technique may be applied to any device-specific data. Additionally, the scope of disclosure is not limited to the boot image contents as described above. The boot images described above may contain any of a variety of contents, such as R&D certificates used for debugging purposes, a primary protected application ("PPA") that is present in secure random access memory after booting, a PPA certificate, and any other appropriate item. Also, while the above subject matter is primarily discussed in terms of applicability to mobile phones, the subject matter may be used with any mobile communication device. Numerous variations and modifications will become apparent to those skilled in the art once the above disclosure is fully appreciated. It is intended that the following claims be interpreted to embrace all such variations and modifications.